(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 February 2001 (01.02.2001)

PCT

(10) International Publication Number
WO 01/08411 A1

(51) International Patent Classification[7]: H04N 7/167

(21) International Application Number: PCT/US00/16982

(22) International Filing Date: 20 June 2000 (20.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/358,687 21 July 1999 (21.07.1999) US

(71) Applicant: CREDENCE SYSTEMS CORPORATION [US/US]; 215 Fourier Avenue, Fremont, CA 94539 (US).

(72) Inventors: BATINIC, Ivan-Pierre; 12215 McConnell Drive, San Martin, CA 95046 (US). KRAUS, Lawrence; 777 N. Second Street, San Jose, CA 95112 (US). LORANGER, Marc, P.; 574 Windermere Circle, Livermore, CA 94550 (US).

(74) Agent: BEDELL, Daniel, J.; Smith-Hill and Bedell, P.C., Suite 104, 12670 N.W. Barnes Road, Portland, OR 97229 (US).

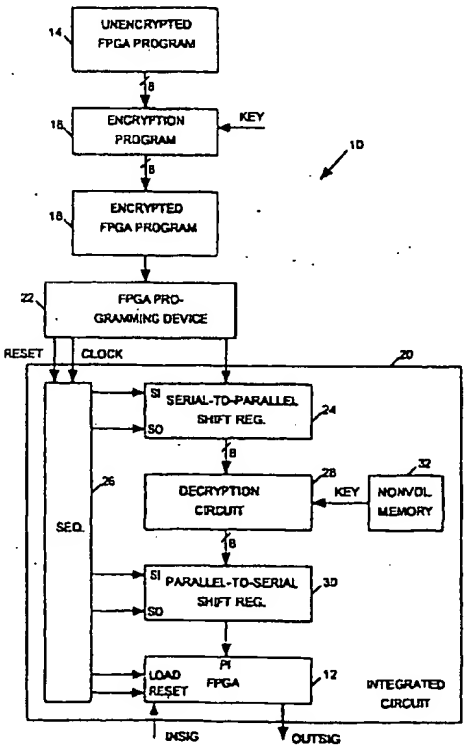
(81) Designated States *(national)*: JP, KR.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:
— *With international search report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: FIELD PROGRAMMABLE GATE ARRAY WITH PROGRAM ENCRYPTION



(57) Abstract: A field programmable gate array (FPGA) and a decryption circuit are implemented within a common integrated circuit (IC) or within separate ICs enclosed within a common IC package. The decryption circuit decrypts an input FPGA program encrypted in accordance with a particular encryption key and then writes the decrypted FPGA program into the FPGA. Thus an FPGA program encrypted in accordance with a particular encryption key can be used to program only those FPGAs coupled with a decryption circuit capable of decoding the encrypted FPGA program in accordance with that particular encryption key. Since the decryption circuit and the FPGA are implemented in the same IC, or within the same IC package, the decrypted FPGA program the decryption circuit produces cannot be readily intercepted and copied.

WO 01/08411 A1

FIELD PROGRAMMABLE GATE ARRAY WITH PROGRAM ENCRYPTION

Background of the Invention

Field of the Invention

The present invention relates in general to a system for encoding a field programmable gate array (FPGA) program so that it can be used to program only a particular set of FPGAs capable of decoding the program.

Description of Related Art

A field programmable gate array (FPGA) includes a large number of logic blocks linked to one another and to FPGA input and output (I/O) terminals through signal routing devices. Routing data stored in the FPGA controls the logic the FPGA carries out on its input signals to produce its output signals by telling the routing devices how to route data between the logic blocks and the FPGA's I/O terminals. An FPGA program is therefore simply a sequence of routing data supplied as input to the FPGA's programming terminals. We typically program (or reprogram) an FPGA by supplying a data file containing the routing data to an FPGA programming device which then writes that routing data into the FPGA.

FPGA program developers or distributors have in interest in preventing unlicensed use of their FPGA programs. For example an FPGA program developer or distributor may find it convenient to allow customers to download FPGA programs from a public internet site, but may want to prevent others from using those programs without first obtaining a license. Or when an FPGA program developer licenses a customer to program only a particular set of FPGAs, the program developer would like to prevent that customer, or anyone else, from using that FPGA program to program other FPGAs.

Some computer programs are encrypted so that they cannot be used to program a computer unless first decrypted in a manner determined by a special code (an "encryption key"). When a customer purchases a license to use the software, the licensor provides the customer with the correct encryption key. When the customer thereafter obtains the software and

circuit then writes the decrypted FPGA program into the FPGA's internal memory array. Thus the encrypted FPGA program can't be used to program FPGAs not equipped with a decryption circuit capable of decrypting the FPGA program in accordance with the appropriate encryption key. Also since the decryption circuit and FPGA are implemented within the same IC, the output terminals of the decryption circuit are not readily accessible. The decrypted FPGA program output of the decryption circuit therefore cannot be not readily intercepted and copied.

In an alternative embodiment of the invention, the decryption circuit and the FPGA are implemented on separate IC chips, but the chips are interconnected and enclosed within the same IC package so that the decrypted FPGA program output of the decryption circuit is not readily accessible.

It is accordingly an object of the invention to provide a system for preventing use of an FPGA program for programming FPGAs other than a particular set of FPGAs for which the FPGA program is intended.

The concluding portion of this specification particularly points out and distinctly claims the subject matter of the present invention. However those skilled in the art will best understand both the organization and method of operation of the invention, together with further advantages and objects thereof, by reading the remaining portions of the specification in view of the accompanying drawing(s) wherein like reference characters refer to like elements.

Brief Description of the Drawing(s)

FIG. 1 illustrates in block diagram form an FPGA program encryption system in accordance with the invention,

FIG. 2 illustrate the encryption program of FIG. 1 in flow chart form,

FIG. 3 illustrates a suitable implementation of the decryption circuit of FIG. 1 in schematic diagram form, and

FIG. 4 illustrates an alternative embodiment of the FPGA program encryption system in accordance with the invention.

Description of the Preferred Embodiment(s)

A field programmable gate array (FPGA) includes a large number of logic blocks linked to one another and to FPGA input and output (I/O) terminals through signal routing devices. Routing data stored in the FPGA controls the logic the FPGA carries out on its input signals to produce its output signals by telling the routing devices how to route data between the logic blocks and the FPGA's I/O terminals. An FPGA "program" is simply a sequence of routing data supplied as input to the FPGA's programming terminals. We typically program (or reprogram) an FPGA by generating a data file containing the routing data and supplying it to a conventional FPGA programming device which then writes that routing data into the FPGA. The present invention relates to an FPGA program encryption system for encrypting an FPGA program so that it can be used to program only a particular set of FPGAs that are adapted to decrypt the program.

FIG. 1 illustrates in block diagram form an FPGA program encryption system 10 in accordance with the invention. System 10 includes a conventional FPGA 12 capable of receiving and storing an FPGA program 14 for controlling logical relationships between the FPGA's input signals (INSIG) and its output signals (OUTSIG). In accordance with the invention, a conventional encryption program 16 encrypts each successive word of FPGA program 14 to produce a corresponding word of an encrypted version 18 of FPGA program 14. Encryption program 16 bases its encryption on an input encryption key (KEY). As illustrated herein, encryption program 16 receives 8-bit FPGA program words, produces 8-bit output encrypted FPGA program words, and employs an 8-bit encryption key. However the key and word size may be larger or smaller in alternative implementations of the system.

FPGA 12 is incorporated into an integrated circuit (IC) 20 along with other circuits (24-32) capable of decrypting encrypted FPGA program 18 and loading the resulting decrypted FPGA program into FPGA 12 via its programming input (PI). A conventional FPGA programming device 22 external to (IC) 20 transmits a RESET signal to a sequencer 26 and then serially

shifts each bit of the encrypted FPGA program into a serial-to-parallel shift register 24, pulsing a CLOCK signal output when each bit is available at the shift register input. Sequencer 26 responds to the RESET signal by pulsing a reset input of FPGA 12 telling it to prepare to receive an input FPGA program. Sequencer 26 thereafter pulses a shift in (SI) input of shift register 24 in response to each pulse of its input CLOCK signal so as to shift each incoming bit of the encoded FPGA program 18 into shift register 24. Shift register 24 assembles its serial input FPGA program into a sequence of 8-bit words and supplies them to a decryption circuit 28. Using the same encryption key (KEY) encryption program 16 used when creating encrypted FPGA program 18, decryption circuit 28 decrypts each of the encrypted program's 8-bit words to reproduce a corresponding 8-bit word of the original unencrypted FPGA program 14. A conventional nonvolatile memory 32 also included in IC 20 stores the encryption KEY and provides it to decryption circuit 28.

Decryption circuit 28 provides each decrypted FPGA program word output as input to a parallel-to-serial shift register 30. Sequencer 26 pulses a shift in (SI) input of shift register 30 to load each successive decrypted FPGA program word into shift register 30 and sequentially pulses its shift out (SO) input causing shift register 30 to serially shift each bit of the word to a programming input (PI) of FPGA 12. Sequencer 26 also sequentially pulses a LOAD input of FPGA 12 telling it to store each bit of the decrypted FPGA program word. Thus devices 24-32 decrypt each word of encrypted FPGA program 18 and write the result into FPGA 12 thereby programming it in accordance with unencrypted FPGA program 14.

Encryption system 10 allows an FPGA program developer or distributor to prevent unlicensed use of FPGA program 14. For example, an FPGA program developer may provide a customer not only the encrypted FPGA program 18, but also an IC 20 including the FPGA 12 to be programmed. Before delivering the IC to the customer, the developer writes the appropriate decryption key into its nonvolatile memory 32. Thus the

customer, or anyone else obtaining a copy of the encrypted FPGA program 18, can use the FPGA program to program only the FPGA 12 included within the particular IC 20 having a decryption circuit 28 capable of appropriately decrypting the FPGA program. Note that since decryption circuit 28 is embedded within IC 20, its output terminals are not readily assessable. Thus a customer cannot readily intercept the decrypted FPGA program it supplies to FPGA 12 and use it to program conventional FPGAs not equipped with the appropriate decryption system. Note also that since the encryption key is also embedded in IC 20, it is not readily available to the customer or anyone else. Thus only the developer will know the key and only the developer will be able to appropriately encrypt programs for IC 20. Therefore when a customer wishes to reprogram FPGA 12 so that it carries out different logic, the customer will have to obtain an appropriately encrypted program from the developer rather than from other sources.

Encryption Program and Decryption Circuit

FIG. 2 illustrates a simple encryption program suitable for use as encryption program 16. Beginning at step 32 encryption program 16 reads a next (first) word of unencrypted program 14 and then exclusive ORs (XORs) each bit of the unencrypted word with a corresponding bit of the encryption key to produce a word of encrypted program 18 output (step 34). When the last processed unencrypted word is not the last word of unencrypted FPGA program 14 (step 36), encryption program 16 repeats steps 32 and 34 to obtain and XOR a next unencrypted word of FPGA program 14 to produce a corresponding word of encrypted program 18. Encryption program 16 ends following step 36 when the last word of unencrypted FPGA program 14 has been encrypted.

FIG. 3 illustrates a simple implementation of decryption circuit 28 of FIG. 1 suitable for decrypting the output of program 16 of FIG. 2. Decryption circuit 28 includes a set of eight exclusive OR (XOR) gates 37. Each XOR gate 37 XORs a bit of an 8-bit encrypted word with a corresponding word of the 8-bit encryption key (KEY) to produce a corresponding

word of decrypted data. Let us assume, for example that an 8-bit word of unencrypted FPGA program 14 has the binary value (10011011) and that the 8-bit encryption key has value (11010010). Encryption program 16 XORs the two 8-bit values to produce a word of encrypted FPGA program 18 having value as follows:

Encrypted = Unencrypted XOR KEY
(01000001) = (10011011) XOR (11010010).

When decryption circuit 28 receives the encrypted word (01000001) and XORs it with the original encryption key value (11010010) it reproduces the original unencrypted word as follows:

Unencrypted = Encrypted XOR KEY
(10011011) = (01000001) XOR (11010010).

An 8-bit encryption key allows for 2^8 different encryption key values. We can increase the size of the encryption key, for example, to 16-bit to allow for 2^{16} possible values. To do so we simply increase to 16 the width of the data word input and output of encryption program 16, the width of the data path between shift register 24 and shift register 30 and the number of gates 37 of decryption circuit 28 of FIG. 3.

Multiple-chip Module Version

FIG. 4 illustrates an alternative embodiment of the invention in which devices 24-32 of FIG. 1 for decrypting the encrypted FPGA program are implemented in an IC chip 38 that does not include FPGA 12. However FPGA 12 and IC chip 38 are mounted within a common IC package 40 using conventional multiple-chip module technology. Since the connection between FPGA 12 and IC chip 38 is within package 40, it is not readily accessible to one wishing to copy the decrypted output sequence produced by IC chip 38.

Design Alternatives

While the forgoing specification has described preferred embodiments of the present invention, one skilled in the art may make many modifications to the preferred embodiment without departing from the invention in its broader aspects. For example, for FPGAs having a programming input (PI) wider than one bit, shift register 30 may be modified to provide a sufficiently wide output. Many well-known systems for encrypting and decrypting data could be used to implement the functions of encryption program 16 and decryption circuit 28 of FIG. 1. Any of several well-known technologies including flash memory cells and fuse technologies could be used to implement nonvolatile memory 32. The encryption key value input to decryption circuit 28 could alternatively be provided as a "hard wired" input to decryption circuit 28 by appropriately programming IC 20 at the mask level during IC fabrication, stored in volatile memory (registers, flip-flops, SRAM etc.) within IC 20. The encryption key may also be provided to decryption circuit 28 from a source external to IC 20. Also while the preferred embodiment of the encryption system is illustrated herein as being used in connection with an FPGA, it should be understood that a similar encryption system could be used in connection with other kinds of logic devices that are programmed by input data sequences. The appended claims are therefore intended to cover all such modifications as fall within the true scope and spirit of the invention.

Claim(s)

What is claimed is:

1. An apparatus for carrying out logic operations on input signals to produce output signals, wherein said logic operations are defined by an encrypted first program supplied as input thereto, said apparatus comprising:

first circuit means for receiving and decrypting said first program to produce a second program, and

second circuit means for receiving and storing said second program and for carrying out logic operations on said input signals to produce said output signals, wherein said logic operations are defined by said second program.

2. The apparatus in accordance with claim 1 wherein said second circuit means comprises a field programmable gate array.

3. The apparatus in accordance with claim 1 wherein said first circuit means and said second circuit means are implemented within in a common integrated circuit.

4. The apparatus in accordance with claim 1 further comprising an integrated circuit package, said first circuit means and said second circuit means being enclosed within said integrated circuit package.

5. The apparatus in accordance with claim 1 further comprising third means for providing said first circuit means with an input encryption key, wherein said first circuit means decrypts said first program in a manner determined by said encryption key.

6. The apparatus in accordance with claim 5 wherein said third means comprises a nonvolitile memory.

7. The apparatus in accordance with claim 1 further comprising:

third means for providing said first circuit means with an input encryption key, wherein said first circuit means decrypts said first program in a manner determined by said encryption key; and

an integrated circuit package, said first circuit means, said second circuit means and said third means being contained within said integrated circuit package.

8. The apparatus in accordance with claim 7 wherein said second circuit means comprises a field programmable gate array.

9. The apparatus in accordance with claim 8 wherein said third means comprises a nonvolitile memory.

10. An apparatus for carrying out logic operations on input signals to produce output signals, wherein said logic operations are defined by a first program supplied as input thereto, said apparatus comprising:

means for encrypting said first program to produce an encrypted second program,

first circuit means for receiving and decrypting said encrypted second program to produce a copy of said first program, and

second circuit means for receiving said copy of said first program and for carrying out logic operations on said input signals to produce said output signals, wherein said logic operations are defined by said copy of said first program.

11. The apparatus in accordance with claim 10 wherein said second circuit means comprises a field programmable gate array.

12. The apparatus in accordance with claim 10 wherein said first circuit means and said second circuit means are implemented within in a common integrated circuit.

13. The apparatus in accordance with claim 10 further comprising an integrated circuit package, said first circuit means and said second circuit means being contained within said integrated circuit package.

14. The apparatus in accordance with claim 10 further comprising third means for providing said first circuit means with an input encryption key, wherein said means for encrypting encrypts said fist program in a manner determined by said encryption key and wherein said first circuit means decrypts said second program in a manner determined by said encryption key.

15. The apparatus in accordance with claim 14 wherein said third means comprises a nonvolitile memory.

16. The apparatus in accordance with claim 14 further comprising an integrated circuit package, said first circuit means and said second circuit means being contained within said integrated circuit package.

17. The apparatus in accordance with claim 14 wherein said first circuit means and said second circuit means are implemented within in a common integrated circuit.

18. A method for programming an apparatus for carrying out logic operations on input signals to produce output signals, wherein said logic operations are defined by a first program supplied as input thereto, said method comprising the steps of:

encrypting said first program in a manner determined by a value of a key to produce an encrypted second program;

decrypting said encrypted second program in a manner determined by said value of said key to produce a copy of said first program, and

supplying said copy of said first program as input to said apparatus.

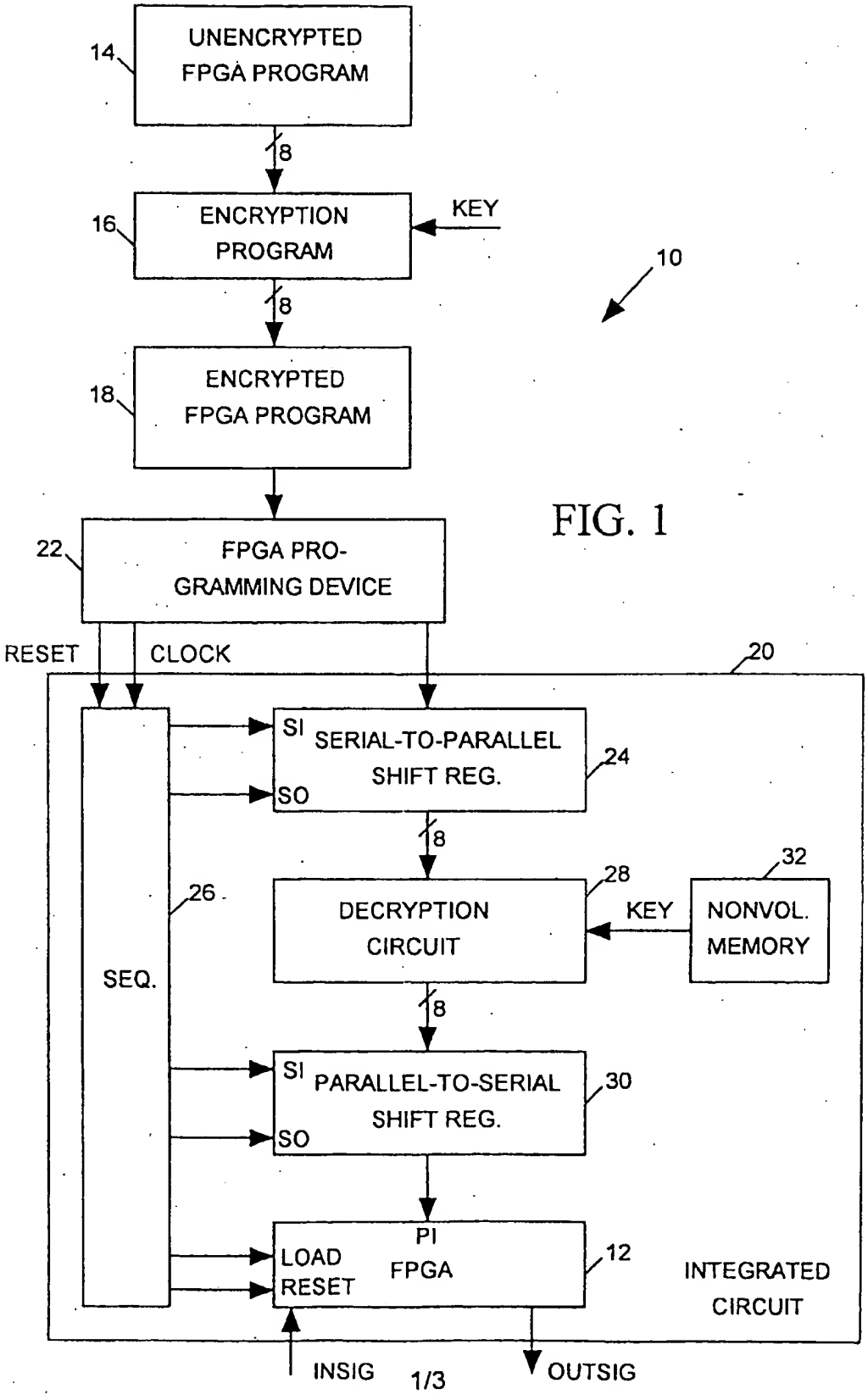

UNENCRYPTED FPGA PROGRAM
14
8
ENCRYPTION PROGRAM
16
KEY
10
8
ENCRYPTED FPGA PROGRAM
18
22
FPGA PRO-GRAMMING DEVICE
RESET
CLOCK
20
SI
SERIAL-TO-PARALLEL SHIFT REG.
SO
24
8
26
28
32
DECRYPTION CIRCUIT
KEY
NONVOL. MEMORY
SEQ.
8
SI
PARALLEL-TO-SERIAL SHIFT REG.
SO
30
PI
LOAD RESET
FPGA
12
INTEGRATED CIRCUIT
INSIG
OUTSIG

FIG. 1

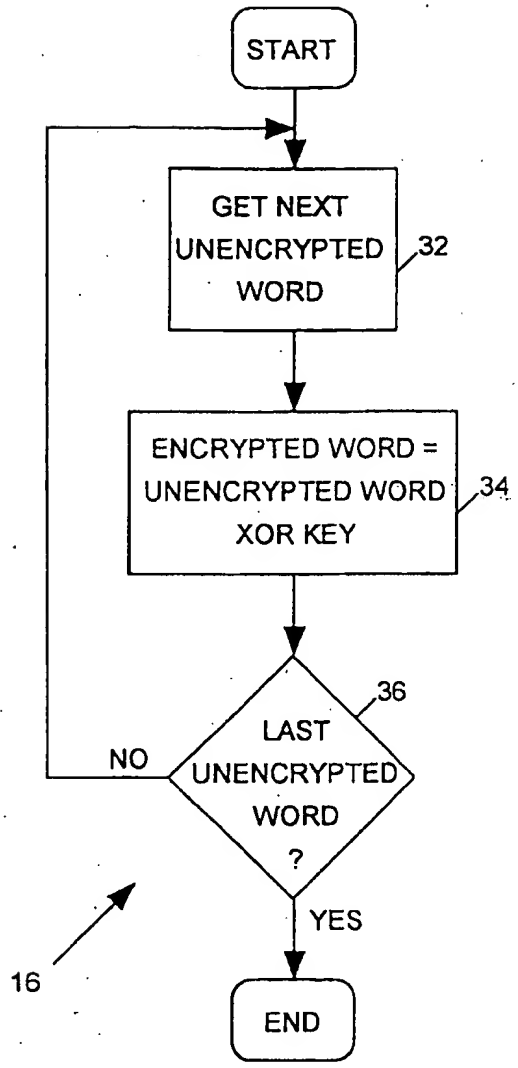

START
GET NEXT UNENCRYPTED WORD
32
ENCRYPTED WORD = UNENCRYPTED WORD XOR KEY
34
LAST UNENCRYPTED WORD ?
36
NO
YES
16
END

FIG. 2

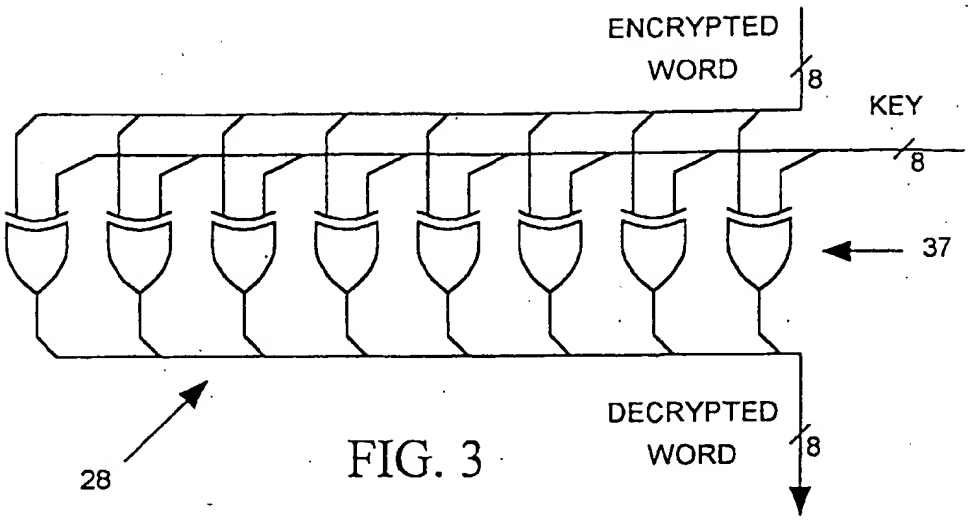

ENCRYPTED
WORD
8
KEY
8
37
DECRYPTED
WORD
8
28

FIG. 3

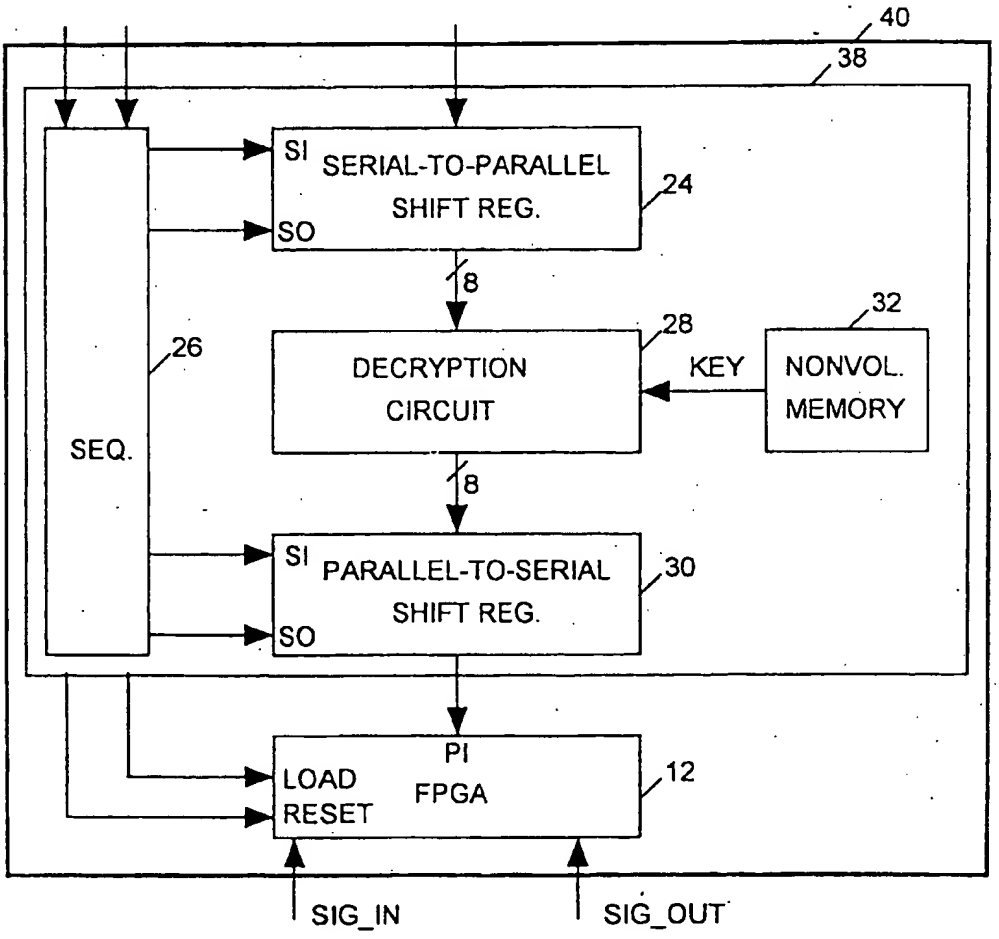

40
38
SI
SERIAL-TO-PARALLEL
SHIFT REG.
SO
24
8
28
32
DECRYPTION
CIRCUIT
KEY
NONVOL.
MEMORY
26
SEQ.
8
SI
PARALLEL-TO-SERIAL
SHIFT REG.
SO
30
PI
LOAD
RESET
FPGA
12
SIG_IN
SIG_OUT

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/16982

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04N 7/167

US CL : 380/264, 283, 713/194, 190, 189

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/264, 283, 713/194, 190, 189

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 6,044,157 A (UESAKA et al.) 28 MARCH 2000, col. 10-17, lines 20-68.	1, 5, 10, 14, 18
Y	US 4,823,308 A (KNIGHT) 18 APRIL 1989, col. 1, lines 55-68.	2-4, 6-8, 11-12, 13, 15-17
Y	US 5,671,281 A (CAMPBELL et al.) 23 September 1997, col. 4-5, lines 22-34.	2-4, 6-8, 11-12, 13, 15-17

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 16 AUGUST 2000	Date of mailing of the international search report 02 OCT 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer TOD SWANN Telephone No. (703) 308-7191

Form PCT/ISA/210 (second sheet) (July 1998)*